

CLAIM AMENDMENTS

- 1 1. (Currently amended) A method of determining a variable to receive a value
2 change and a value for the variable as part of a local search solution to an integer
3 programming problem having polynomial terms of at least second order
4 comprising the steps of:
 - 5 ~~a.~~ selecting an unsatisfied constraint;
 - 6 ~~b.~~ creating stores for allowable changes of value for the variables in the
7 unsatisfied constraint;
 - 8 ~~c.~~ parsing through the unsatisfied constraint by term and for each variable in
9 the term updating the stores with a change in the term for each of the
10 allowable changes of the value while maintaining other variables constant; and
 - 11 ~~d.~~ choosing the variable to receive the value change and the value for the
12 variable based upon the store which meets an improvement criterion.
- 1 2. (Original) The method of claim 1 wherein the variables comprise a binary
2 variable.
- 1 3. (Original) The method of claim 2 wherein all of the variables comprise
2 binary variables.
- 1 4. (Original) The method of claim 3 wherein each of the variables is
2 associated with no more than a single store.
- 1 5. (Original) The method of claim 3 wherein the step of choosing the
2 variable to receive the value change and the value for the variable comprises
3 choosing the variables to receive the value change based upon which of the stores
4 meets the improvement criterion and flipping the value of the variable selected.
- 1 6. (Original) The method of claim 1 wherein the allowable values for at least
2 one of the variables comprise integer values.
- 1 7. (Original) The method of claim 6 wherein the integer values comprise a
2 subset of possible integer values.

1 8. (Original) The method of claim 7 wherein the subset of the possible
2 integer values comprises the integer values near an existing value for the variable.

1 9. (Original) The method of claim 1 wherein the allowable changes in the
2 variables do not include tabu values.

1 10. (Original) The method of claim 1 wherein the improvement criterion
2 comprises the store which improves the unsatisfied constraint and which improves
3 an overall solution at least as much as other stores.

1 11. (Original) The method of claim 1 wherein the improvement criterion
2 comprises the store which is not tabu, which improves the unsatisfied constraint,
3 and which improves an overall solution at least as much as other stores which are
4 not tabu.

1 12. (Original) The method of claim 1 further comprising the step of
2 determining that none of the stores improve an overall solution.

1 13. (Original) The method of claim 12 wherein the improvement criterion
2 comprises the store which improves the unsatisfied constraint at least as much as
3 other stores.

1 14. (Original) The method of claim 12 wherein the improvement criterion
2 comprises a random selection of one of the stores which improves the unsatisfied
3 constraint.

1 15. (Original) The method of claim 1 wherein the improvement criterion
2 comprises improvement criteria.

1 16. (Currently amended) The method of claim 1 further comprising the steps of:
2 a. defining a problem model which comprises constraints, wherein the
3 constraints comprise hard constraints and a soft constraint;
4 b. randomly initializing values for the variables, thereby forming assigned

5 values for the variables, wherein the assigned values are applied to the
6 constraints to determine the unsatisfied constraints;
7 e- changing the value of the variable according to the improvement criterion,
8 thereby modifying the assigned values to form new assigned values; and
9 d- comparing the constraints to optimality criteria and if the optimality
10 criteria are met, outputting the new assigned values as a near optimum
11 solution.

1 17. (Currently amended) The method of claim 16 wherein, if the optimality
2 criteria are not met, further comprising determining whether to perform an
3 additional iteration and if so, returning to the step e of changing the value of the
4 variable, wherein the new assigned values become the assigned values.

1 18. (Currently amended) The method of claim 17 wherein, if another iteration is
2 not to be performed, further comprising determining whether to reinitialize the
3 variables and if so, returning to the step b of randomly initializing the values for
4 the variables.

1 19. (Original) The method of claim 18 wherein, if the variables are not to be
2 reinitialized, outputting a no solution found message.

1 20. (Currently amended) A method of determining a binary variable to receive a
2 value change as part of a solution to an integer programming problem having
3 polynomial terms of at least second order comprising the steps of:
4 a- selecting an unsatisfied constraint;
5 b- creating a store for each binary variable in the unsatisfied constraint;
6 c- parsing through the unsatisfied constraint by term and for each binary
7 variable in the term updating the store for the binary variable with a change in
8 the term due to flipping a value of the binary variable while maintaining other
9 variables constant; and
10 d- choosing the binary variable to receive the value change based upon the
11 store which meets an improvement criterion.

1 21. (Currently amended) A method of determining a variable to receive a value

2 change and a value for the variable as part of a local search solution to an integer
3 programming problem having polynomial terms of at least second order
4 comprising the steps of:

- 5 a- selecting an unsatisfied constraint;
- 6 b- creating stores for allowable changes of value for the variables in the
7 unsatisfied constraint;
- 8 c- parsing through the unsatisfied constraint by term and for each variable in
9 the term which is encountered for a first time updating the stores with a
10 change in the unsatisfied constraint for each of the allowable changes of the
11 value while maintaining other variables constant; and
- 12 d- choosing the variable to receive the value change and the value for the
13 variable based upon the store which meets an improvement criterion.

1 22. (Currently amended) A computer readable memory comprising computer
2 code for directing a computer to make a determination of a variable to receive a
3 value change and a value for the variable as part of a solution to an integer
4 programming problem having polynomial terms of at least second order, the
5 determination of the variable to change and the value comprising the steps of:
6 a- selecting an unsatisfied constraint;
7 b- creating stores in memory for allowable changes of value for the variables
8 in the unsatisfied constraint;
9 c- parsing through the unsatisfied constraint by term and for each variable in
10 the term updating the stores with a change in the term for each of the
11 allowable changes of the value while maintaining other variables constant; and
12 d- choosing the variable to receive the value change and the value for the
13 variable based upon the store which meets an improvement criterion.

1 23. (Original) The computer readable memory of claim 22 wherein the
2 variables comprise a binary variable.

1 24. (Original) The computer readable memory of claim 23 wherein all of the
2 variables comprise binary variables.

1 25. (Original) The computer readable memory of claim 24 wherein each of

2 the variables is associated with no more than a single store.

1 26. (Original) The computer readable memory of claim 25 wherein the step of
2 choosing the variable to receive the value change and the value for the variable
3 comprises choosing the variable to receive the value change based upon which of
4 the stores meets the improvement criterion and flipping the value of the variable
5 selected.

1 27. (Original) The computer readable memory of claim 22 wherein the
2 allowable values for at least one of the variables comprise integer values.

1 28. (Original) The computer readable memory of claim 27 wherein the integer
2 values comprise a subset of possible integer values.

1 29. (Original) The computer readable memory of claim 28 wherein the subset
2 of the possible integer values comprises the integer values near an existing value
3 for the variable.

1 30. (Original) The computer readable memory of claim 22 wherein the
2 allowable changes in the variables do not include tabu values.

1 31. (Original) The computer readable memory of claim 22 wherein the
2 improvement criterion comprises the store which improves the unsatisfied
3 constraint and improves an overall solution at least as much as other stores.

1 32. (Original) The computer readable memory of claim 22 wherein the
2 improvement criterion comprises the store which is not tabu, which improves the
3 unsatisfied constraint, and which improves an overall solution at least as much as
4 other stores which are not tabu.

1 33. (Original) The computer readable memory of claim 22 further comprising
2 the step of determining that none of the stores improve an overall solution.

1 34. (Original) The computer readable memory of claim 33 wherein the

2 improvement criterion comprises the store which improves the unsatisfied
3 constraint at least as much as other stores.

1 35. (Original) The computer readable memory of claim 33 wherein the
2 improvement criterion comprises a random selection of one of the stores which
3 improves the unsatisfied constraint.

1 36. (Original) The computer readable memory of claim 22 wherein the
2 improvement criterion comprises improvement criteria.

1 37. (Currently amended) The computer readable memory of claim 22 further
2 comprising the steps of:
3 a. defining a problem model which comprises constraints, wherein the
4 constraints comprise hard constraints and a soft constraint;
5 b. randomly initializing values for the variables, thereby forming assigned
6 values for the variables, wherein the assigned values are applied to the
7 constraints to determine the unsatisfied constraints;
8 c. changing the value of the variable according to the improvement criterion,
9 thereby modifying the assigned values to form new assigned values;
10 d. comparing the constraints to optimality criteria and if the optimality
11 criteria are met, outputting the new assigned values as a near optimum
12 solution.

1 38. (Currently amended) The computer readable memory of claim 37 wherein, if
2 the optimality criteria are not met, further comprising determining whether to
3 perform an additional iteration and if so, returning to the step c of changing the
4 value of the variable, wherein the new assigned values become the assigned
5 values.

1 39. (Currently amended) The computer readable memory of claim 38 wherein, if
2 another iteration is not to be performed, further comprising determining whether
3 to reinitialize the variables and if so, returning to the step b of randomly
4 initializing the values for the variables.

1 40. (Original) The computer readable memory of claim 39 wherein, if the
2 variables are not to be reinitialized, outputting a no solution found message.

1 41. (Currently amended) A computer readable memory comprising computer
2 code for directing a computer to make a determination of a binary variable to
3 receive a value change as part of a solution to an integer programming problem
4 having polynomial terms of at least second order, the determination of the binary
5 variable comprising the steps of:

6 a. selecting an unsatisfied constraint;

7 b. creating a store in memory for each binary variable in the unsatisfied
8 constraint;

9 c. parsing through the unsatisfied constraint by term and for each binary
10 variable in the term updating the store for the binary variable with a change in
11 the term due to flipping a value of the binary variable while maintaining other
12 variables constant; and

13 d. choosing the binary variable to receive the value change based upon the
14 store which meets an improvement criterion.

1 42. (Currently amended) A computer readable memory comprising computer
2 code for directing a computer to make a determination of a variable to receive a
3 value change and a value for the variable as part of a solution to an integer
4 programming problem having polynomial terms of at least second order, the
5 determination of the variable to change and the value comprising the steps of:

6 a. selecting an unsatisfied constraint;

7 b. creating stores in memory for allowable changes of value for the variables
8 in the unsatisfied constraint;

9 c. parsing through the unsatisfied constraint by term and for each variable in
10 the term which is encountered for a first time updating the stores with a
11 change in the unsatisfied constraint for each of the allowable changes of the
12 value while maintaining other variables constant; and

13 d. choosing the variable to receive the value change and the value for the
14 variable based upon the store which meets an improvement criterion.